

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Hyun-Kwon CHUNG et al

Application No.: Unassigned

Group Art Unit: Unassigned

Filed: November 20, 2003

Examiner: Unassigned

For: METHOD AND APPARATUS FOR DISPLAYING MARKUP DOCUMENT LINKED TO
APPLET

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith
a certified copy of the following foreign application:

Korean Patent Application No(s). 2002-79185

Filed: December 12, 2002

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

By: 

Michael D. Stein
Registration No. 37,240

Date: November 20, 2003

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 10-2002-0079185
Application Number

출원 년 월 일 : 2002년 12월 12일
Date of Application DEC 12, 2002

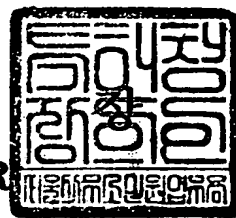
출원인 : 삼성전자주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.



2003 년 07 월 29 일

특 허 청

COMMISSIONER



【서지사항】

·【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0020
【제출일자】	2002. 12. 12
【국제특허분류】	G06F
【발명의 명칭】	애플릿이 링크된 마크업 문서의 디스플레이 방법 및 그 시스템
【발명의 영문명칭】	Method and apparatus for displaying of markup document linking applet
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이영필
【대리인코드】	9-1998-000334-6
【포괄위임등록번호】	1999-009556-9
【대리인】	
【성명】	이해영
【대리인코드】	9-1999-000227-4
【포괄위임등록번호】	2000-002816-9
【발명자】	
【성명의 국문표기】	정현권
【성명의 영문표기】	CHUNG, Hyun Kwon
【주민등록번호】	721217-1042731
【우편번호】	464-800
【주소】	경기도 광주군 광주읍 탄벌리 동보아파트 104동 906호
【국적】	KR
【발명자】	
【성명의 국문표기】	허정권
【성명의 영문표기】	HEO, Jung Kwon
【주민등록번호】	681207-1830616



1020020079185

출력 일자: 2003/7/29

【우편번호】	137-766
【주소】	서울특별시 서초구 반포2동 주공아파트 2단지 203동 504호
【국적】	KR
【발명자】	
【성명의 국문표기】	박성욱
【성명의 영문표기】	PARK, Sung Wook
【주민등록번호】	710327-1041719
【우편번호】	137-073
【주소】	서울특별시 서초구 서초3동 1595-2 센쥬리오피스텔 2동 1207호
【국적】	KR
【발명자】	
【성명의 국문표기】	정길수
【성명의 영문표기】	JUNG, Kil Soo
【주민등록번호】	750903-1917317
【우편번호】	445-970
【주소】	경기도 화성군 태안읍 병점 한신아파트 107동 707호
【국적】	KR
【취지】	특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대리인 이영필 (인) 대리인 이해영 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	2 면 2,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	0 항 0 원
【합계】	31,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

애플릿이 링크된 마크업 문서의 디스플레이 방법 및 그 장치가 개시된다.

본 발명에 따라 애플릿이 링크된 마크업 문서를 디스플레이하는 방법은 (a) 상기 마크업 문서를 위한 화면 출력 정보의 디스플레이를 지연시키는 단계; 및 (b) 상기 애플릿의 렌더링이 완료되면 지연된 화면 출력 정보와 상기 애플릿 출력을 동기시켜 디스플레이하는 단계를 포함하는 것을 특징으로 한다. 이에 의해, 애플릿과 마크업 문서를 동기시켜 디스플레이할 수 있으므로, 애플릿이 초기화되고 실행 시작을 준비하는 동안에 종래 사용자에게 보여지던 빈 화면이 더 이상 보이지 않게 되어, 사용자에게 보다 편안하게 화면을 제공할 수 있게 된다.

【대표도】

도 2

【명세서】

【발명의 명칭】

애플릿이 링크된 마크업 문서의 디스플레이 방법 및 그 시스템{Method and apparatus for displaying of markup document linking applet}

【도면의 간단한 설명】

도 1은 본 발명의 바람직한 실시예에 따른 시스템(100)의 블록도,

도 2는 본 발명의 일 실시예에 따른 프리젠테이션 엔진(1)의 블록도,

도 3은 본 발명의 바람직한 실시예에 따른 디스플레이 방법을 설명하기 위한 디코더(17)의 상세 블록도,

도 4는 본 발명에 따른 애플릿의 동작 방식을 설명하기 위한 참고도,

도 5는 본 발명에 따라 디스플레이되는, 애플릿을 포함한 마크업 화면의 일 예,

도 6은 종래 자바 애플릿이 링크된 마크업 문서가 디스플레이되는 초기 화면의 일 예이다.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<7> 본 발명은 마크업 문서의 디스플레이에 관한 것으로, 보다 상세하게는 애플릿을 포함하는 마크업 문서의 디스플레이 방법 및 그 장치에 관한 것이다.

<8> 사용자와의 인터랙션을 원활하게 수행하거나 보다 동적인 웹페이지를 구성하기 위해 자바 애플릿(applet)이 링크되어 있는 마크업 문서가 많이 제작되고 있다. 마크업 문서는 HTML,

XML 등 마크업 언어로 작성된 문서로서 텍스트, 이미지, 그래픽, 등 마크업 리소스를 보여주는 콘텐츠임과 동시에 마크업 언어를 해석할 수 있는 해석 엔진에 의해 실행되는 어플리케이션 역할을 수행한다.

<9> 도 6은 종래 자바 애플릿이 링크된 마크업 문서가 디스플레이되는 초기 화면의 일 예이다.

<10> 도 6을 참조하면, 사용자가 해당 마크업 문서를 불러와서 실행시키면 마크업 문서의 외곽을 구성하는 화면이 먼저 디스플레이되고 자바 애플릿은 나중에 디스플레이된다. 이는 자바 애플릿이 초기화되고 실행이 시작되어 그 첫번째 화면이 디스플레이되기까지 다소 시간이 소요되기 때문이다. 따라서, 사용자는 자바 애플릿의 첫번째 화면이 디스플레이될 때까지 1) 아무것도 표시되어 있지 않거나 2) 자바 애플릿을 실행하기 위한 대응 클래스를 로딩하고 있음을 알리는 메시지만 표시되거나 3) 초기화를 수행하고 있음을 알리는 메시지만 표시된, 비어 있는 애플릿 영역을 가진 마크업 화면을 보아야 한다.

【발명이 이루고자 하는 기술적 과제】

<11> 따라서, 본 발명의 목적은 애플릿이 링크된 마크업 문서가 디스플레이되는 경우 애플릿과 마크업 문서가 동시에 디스플레이되도록, 애플릿이 링크된 마크업 문서를 디스플레이하는 방법 및 그 시스템을 제공하는 것이다.

【발명의 구성 및 작용】

<12> 상기 목적은 본 발명에 따라, 애플릿이 링크된 마크업 문서의 디스플레이 방법에 있어서, (a) 상기 마크업 문서를 위한 화면 출력 정보의 디스플레이를 지연시키는 단계; 및 (b) 상기

애플릿의 렌더링이 완료되면 지연된 화면 출력 정보와 상기 애플릿 출력을 동기시켜 디스플레이하는 단계를 포함하는 것을 특징으로 하는 방법에 의해 달성된다.

- <13> 상기 (a)단계는 상기 마크업 문서를 위한 화면 출력 정보를 버퍼링하는 단계를 포함하는 것이 바람직하다.
- <14> 상기 (b)단계는 출력 제어 신호에 기초하여 상기 지연된 화면 출력 정보와 상기 애플릿의 첫번째 화면을 위한 애플릿 출력을 디스플레이 장치로 동기시켜 제공하는 단계를 포함하는 것이 바람직하다.
- <15> 상기 애플릿은 마크업 문서의 화면 출력과는 별개의 출력 방식을 가진 프로그램 코드로 이루어진 것이 바람직하다.
- <16> 상기 출력 제어 신호는 상기 애플릿을 해석하기 위한 애플릿 실행 엔진 또는 상기 마크업 문서를 해석하기 위한 프리젠테이션 엔진으로부터 제공됨이 바람직하다.
- <17> 상기 (a)단계는 상기 화면 출력 정보로서 텍스트 출력을 포함하고, 이미지 출력 및 오디오 출력 중 적어도 하나를 더 포함하는 것이 바람직하다.
- <18> 한편, 본 발명의 다른 분야에 따르면, 상기 목적은 마크업 문서; 및 상기 마크업 문서에 링크된 애플릿을 포함하고, 상기 애플릿 또는 상기 마크업 문서는 상기 마크업 문서의 디스플레이를 지연시키는 화면 출력 지연 정보를 포함하는 것을 특징으로 하는 정보저장매체에 의해서도 달성된다.
- <19> 상기 애플릿은 상기 마크업 문서를 디스플레이하기 위한 화면 출력 정보와 상기 애플릿의 출력 정보를 동기화 시키기 위한 디스플레이 지연 정보로서 지연 함수를 포함하는 것이 바

람직하고, 초기 상태, 실행 상태, 종료 상태 및 삭제 상태 중 어느 하나의 상태에서 동작하는 것이 바람직하다.

<20> 상기 애플릿은 상기 마크업 문서를 위한 화면 출력 정보의 디스플레이를 지연시키는 지연 정보로서 지연 함수; 및 상기 애플릿이 초기화되고 실행이 시작되어 그 첫번째 화면의 렌더링이 완료되면 상기 마크업 문서의 디스플레이의 지연을 해제하는 지연 해제 함수를 포함하거나, 상기 마크업 문서는 상기 애플릿의 출력 정보와 동기화되어 디스플레이될 것을 요구하는 태그(tag) 또는 속성(attribute) 지시 정보를 포함하는 것이 바람직하다.

<21> 한편, 본 발명의 다른 분야에 따르면, 상기 목적은 마크업 문서를 해석하는 프리젠테이션 엔진; 및 상기 마크업 문서에 링크된 애플릿을 해석하는 애플릿 실행 엔진을 포함하고, 상기 프리젠테이션 엔진은 상기 마크업 문서를 위한 화면 출력 정보의 디스플레이를 지연시키고, 상기 애플릿 실행 엔진으로부터 상기 애플릿의 출력에 대한 렌더링이 완료되었음을 알리는 출력 제어 신호가 입력되면, 지연된 화면 출력 정보와 상기 애플릿의 출력을 동기시켜 디스플레이 장치로 출력하는 것을 특징으로 하는 시스템에 의해서도 달성된다.

<22> 상기 프리젠테이션 엔진은 상기 마크업 문서를 위한 화면 출력 정보의 디스플레이를 지연시키기 위해 버퍼링하는 버퍼를 포함하는 것이 바람직하다.

<23> 상기 프리젠테이션 엔진은 오디오 출력을 버퍼링하기 위한 오디오 버퍼와 비디오 출력을 버퍼링하기 위한 비디오 버퍼를 구비함이 바람직하다.

<24> 이하에서는 첨부된 도면을 참조하여 본 발명의 바람직한 실시예에 대해 상세히 설명한다

<25> 도 1은 본 발명의 바람직한 실시예에 따른 시스템(100)의 블록도이다.

- <26> 도 1을 참조하면, 시스템(100)은 인터넷 등의 네트워크에 연결가능하고 광 디스크, 자기 디스크 등 미디어에 연결가능한 시스템으로서, 프로세서(5)와 스토리지(4)를 구비한다.
- <27> 스토리지(4)에는 운영체제(3)가 탑재되어 있고, 운영체제(3)와 인터페이스하는 프리젠테이션 엔진(1)과 애플릿 실행 엔진(2)이 탑재되어 있다. 이들은 프로세서(5)에 의해 호출되어 실행된다. 프리젠테이션 엔진(1)은 마크업 문서를 해석하는 엔진으로, 본 발명의 디스플레이 방법에 따라 애플릿과 마크업 문서가 동기되어 디스플레이되도록 애플릿 또는 마크업 문서의 렌더링 및 디스플레이 장치(도시되지 않음)로의 출력을 제어한다. 애플릿 실행 엔진(2)은 마크업 문서에 링크된 애플릿을 해석하는 엔진으로, 본 실시예에 따라 제작된 자바 애플릿을 해석하여 프리젠테이션 엔진(1)으로 출력 제어 신호를 제공한다. 자바 애플릿은 Java 프로그램 언어로 작성된 작은 규모의 어플리케이션으로, 마크업 문서의 화면 출력과는 별개의 출력 방식을 가진 프로그램 코드로 이루어진다.
- <28> 도 2는 본 발명의 일 실시예에 따른 프리젠테이션 엔진(1)의 블록도이다.
- <29> 도 2를 참조하면, 프리젠테이션 엔진(1)은 마크업 문서 파서(11), 스타일시트 파서(12), 스크립트 인터프리터(13), 돔(DOM: Document Object Model) 매니플레이터(14), UI 컨트롤러(15), 레이아웃 포맷터(16) 및 디코더(17)를 포함한다.
- <30> 마크업 문서 파서(11)는 본 실시예에 따라 XHTML 언어로 작성된 XHTML 문서를 파싱하여 문서 트리를 생성하기 위한 정보를 DOM 매니플레이터(14)로 전달한다. 스타일시트 파서(12)는 스타일시트를 해석하여 스타일 규칙 및 셀렉터 리스트(Style rule/Selector list)를 생성하기 위한 정보를 DOM 매니플레이터(14)로 전달한다. 스타일시트는 마크업 문서의 양식을 자유롭게 지정할 수 있게 해준다. 본 실시예에서 스타일시트의 문법과 형식은 W3C의 Cascading Style Sheet(CSS)에 따른다.

<31> 스크립트 인터프리터(13)는 마크업 문서에 포함된 스크립트 코드를 해석하여 그 결과를 DOM 매니플레이터(14)로 전달한다. DOM 매니플레이터(14)는 마크업 문서를 프로그램 객체로 만들거나 수정할 수 있도록 해준다. 즉, 스크립트 인터프리터(13)와의 상호작용에 따라 문서 트리, 및 스타일 규칙/셀렉터 리스트를 수정하거나 보완한다. 또한 DOM 매니플레이터(14)는 마크업 문서 파서(11)에서 입력받은 문서 트리를 재분석하고 실행해야 할 스크립트 코드가 연결되어 있으면 스크립트 인터프리터(13)로 알려주고 스크립트 인터프리터(13)는 이를 실행한다. 또한, DOM 매니플레이터(14)는 마크업 문서에 링크된 이미지, 오디오 등 마크업 리소스를 프리젠테이션 엔진(1)의 메모리(도시되지 않음) 상으로 읽어들이며, 마크업 문서에 링크된 애플릿을 애플릿 실행 엔진(2)의 메모리(도시되지 않음)에 로드하고 애플릿을 초기화하기 위한 초기화 함수 `init()`를 호출하여 실행한다.

<32> 레이아웃 포맷터(16)는 스타일 규칙/셀렉터 리스트를 문서 트리에 적용하고 이를 기초로 출력되는 문서의 양식(예를 들면, 컴퓨터 화면에 출력될 페이지인지 TV 화면에 출력될 페이지인지, 사운드인지, 등)에 따라 대응 포매팅 구조(formatting structure)를 생성한다. 포매팅 구조는 문서 트리과 일견 유사한 구조를 가진 것처럼 보이지만 의사-엘리먼트(pseudo-element)를 활용할 수 있으며 반드시 트리 구조를 갖지 않아도 된다. 즉, 구현(implementation)에 종속적이다. 또한, 포매팅 구조는 문서 트리보다 많은 정보를 가질 수도 있고 그 반대일 수도 있다. 예를 들어, 문서 트리의 엘리먼트가 'display' 속성값으로 'none' 값을 가지면, 그 엘리먼트는 포매팅 구조를 위한 어떤값도 생성하지 않는다. 본 실시예에서 포매팅 구조는 CSS(Cascading Style Sheet) 2 프로세싱 모델에 따른다. 포매팅 구조에 대한 보다 상세한 설명은 CSS 2 프로세싱 모델에 개시되어 있다. 레이아웃 포맷터(16)는 생성된 포매팅 구조를,

출력되는 문서의 양식(즉, 타겟 매체)에 따라 마크업 문서를 렌더링하기 위한 디코더(17)로 보낸다.

<33> 디코더(17)는 마크업 문서를 디스플레이하기 위한 이미지, 그래픽, 오디오 등 마크업 리소스(binary content resource)를 디코딩한 다음 화면에 요구되는 위치에 출력되도록 렌더링하는 한편, 애플릿 실행 엔진(2)으로부터 애플릿의 첫번째 화면을 위한 애플릿 출력을 제공받아 요구되는 위치에 출력되도록 렌더링한다. 마크업 리소스의 렌더링과 애플릿 출력의 렌더링의 실행 순서는 상황에 따라 달라질 수 있다. 예를 들어, 마크업 리소스가 많아서 렌더링에 보다 많은 시간이 소요되는 경우 디코더(17)는 마크업 리소스의 렌더링을 먼저 수행할 수 있다.

<34> 애플릿 실행 엔진(2)은 마크업 문서에 링크된, 본 발명에 따른 화면 출력 지연 정보가 포함된 애플릿을 해석하여 출력 제어 신호를 디코더(17)로 출력하고, 이에 따라 디코더(17)는

- 1) 마크업 문서의 화면 출력을 위한 렌더링을 잠시 보류하였다가 애플릿을 먼저 렌더링하거나
- 2) 마크업 문서의 렌더링은 완료하였으나 디스플레이 장치로 출력하지 않고 애플릿의 렌더링이 완료될 때까지 기다렸다가 마크업 문서와 애플릿의 첫번째 화면을 함께 디스플레이 장치로 출력함으로써 마크업 문서로부터 얻어진 마크업 화면과 애플릿의 첫번째 화면이 동시에 디스플레이 되도록 한다.

<35> 도 3은 본 발명의 바람직한 실시예에 따른 디스플레이 방법을 설명하기 위한 디코더(17)의 상세 블록도이다.

<36> 도 3을 참조하면, 디코더(17)는 본 실시예에 따라 2 개의 버퍼, 제1 디코더 버퍼(171) 및 제2 디코더 버퍼(172)를 구비한다. 제1 디코더 버퍼(171)는 비디오 데이터를 버퍼링하기 위한 버퍼이고, 제2 디코더 버퍼(172)는 오디오 데이터를 버퍼링하기 위한 버퍼이다.

- <37> 디코더(17)는 레이아웃 포맷터(16)로부터 출력된, 화면 출력 정보를 입력받는 한편, 디코딩해야 할 마크업 리소스가 있으면 디코딩한다. 화면 출력 정보는 가령, 마크업 문서로부터의 텍스트, 마크업 문서에 링크된 애플릿, 이미지 및 오디오로 구성된다. 여기서, 애플릿 출력은 애플릿 실행 엔진(2)으로부터 제공되는 것으로, 이미지 출력 및 오디오 출력 중 적어도 하나를 포함할 수 있다. 텍스트 출력, 이미지 출력, 애플릿 출력의 이미지 출력, 텍스트 출력 및 이미지 출력은 제1 디코더 버퍼(171)로 입력되어 버퍼링되고 애플릿 출력의 오디오 출력 및 오디오 출력은 제2 디코더 버퍼(172)로 입력되어 버퍼링된다.
- <38> 애플릿 실행 엔진(2)은 본 발명에 따라 작성된 애플릿을 해석하고 실행하여, 애플릿의 화면 출력 지연 정보에 따른 출력 제어 신호를 디코더(17)로 전달한다.
- <39> 디코더(17)는 애플릿이 실행되어 그 첫번째 화면이 렌더링될 때까지, 디코딩된 나머지 화면 출력 정보가 디스플레이 장치(200)로 입력되지 않고 디코더 버퍼, 즉 제1 디코더 버퍼(171)와 제2 디코더 버퍼(172)에 버퍼링되도록 한 다음 애플릿 실행 엔진(2)으로부터 출력 제어 신호에 따라 애플릿 출력 및 나머지 화면 출력 정보가 동시에 디스플레이 장치(200)로 입력되도록 스위치(173)를 제어한다.
- <40> 다시 말해, 애플릿 실행 엔진(2)은 자바 애플릿을 초기화하고 실행을 시작하여 디스플레이 가능한 상태가 되면 ResumeRender 신호를 디코더(17)로 보내주고, 이에 디코더(17)의 버퍼와 디스플레이 장치(200)의 버퍼가 연결된다. 즉, 제1 디코더 버퍼(171), 제2 디코더 버퍼(172)와 제1 디스플레이 버퍼(201), 제2 디스플레이 버퍼(202)가 각각 연결된다.
- <41> ResumeRender 신호는 본 발명에 따른 출력 제어 신호로서, 애플릿 실행 엔진(2)은 자바 애플릿에 포함된 SuspendRender()를 호출하거나, 또는 마크업 문서에 포함된 태그(tag) 또는 속성(attribute)에 따라 발생될 수 있다. ResumeRender 신호가 발생되면 나머지 화면 출력 정

보들이 디스플레이 장치(200)로 입력되지 않고 디코더(17)의 버퍼에 버퍼링되도록 한 다음 자바 애플릿이 실행되고 그 첫번째 화면이 디스플레이될 수 있도록 애플릿 실행 엔진(2)은 애플릿의 paint()를 호출하여 렌더링을 완료하고 나서 ResumeRender 신호를 발생시켜 제1 디코더 버퍼(171), 제2 디코더 버퍼(172)와 제1 디스플레이 버퍼(201), 제2 디스플레이 버퍼(202)가 각각 연결되도록 한다. 버퍼들이 연결되면 출력이 지연되었던 마크업 화면과, 애플릿의 첫번째 화면이 함께 디스플레이 장치(200)로 출력되어 동시에 디스플레이된다.

<42> 한편, 본 실시예에서 ResumeRender 신호는 애플릿 실행 엔진(2)에서 생성하였으나 프리젠테이션 엔진(1)이 생성하도록 구성할 수 있다.

<43> 도 4는 본 발명에 따른 애플릿의 동작 방식을 설명하기 위한 참고도이다.

<44> 도 4를 참조하면, 프리젠테이션 엔진(1)이 DOM 트리의 생성을 완료하자마자 애플릿이 애플릿 실행 엔진(2)의 메모리(도시되지 않음)에 로드되고 애플릿 실행 엔진(2)은 초기화를 위해 init()를 호출한다. 이에, 애플릿은 초기화 상태(Init)가 된다. 다음으로, 애플릿 실행 엔진(2)은 애플릿의 시작을 위해 start()를 호출한다. 이에, 애플릿의 실행이 시작되는 실행 상태(Start)가 된다. 실행 상태에서 애플릿의 화면들이 디스플레이된다. 애플릿 화면이 디스플레이되고 사용자와의 인터랙션이 수행되는 동안 언로드 이벤트가 발생되면 애플릿 실행 엔진(2)은 stop()을 호출한다. 언로드 이벤트는 예를 들어 사용자가 다른 마크업 화면으로 이동하기 위한 버튼을 누르는 등 현재 마크업 화면의 디스플레이를 종료하기 위한 이벤트를 말한다. stop()이 호출되면 애플릿의 디스플레이를 종료하는 종료 상태(Stop)로 천이한다. 만약, 디스플레이가 종료된 마크업 화면에 대한 사용자의 재방문이 이루어지면 애플릿 실행 엔진(2)은 다시 start()를 호출하고 이에 애플릿의 실행이 다시 시작되는 실행 상태(Start)로 천이한다. 한편, 애플릿이 메모리로부터 삭제되면 애플릿이 완전히 사라지는 삭제 상태(Destroy)가 된다.

<45> 다음은 마크업 화면에 포함된 자바 애플릿의 하나인 애니메이션 애플릿을 위한 소스 코드의 일 예이다.

```
<46> public class AnimationApplet extends Applet
        implements Runnable {
    Thread animator;
    Animation animation;

    public void init() {
        animator = new Thread(this);
        animation = new RectangleAnimation();
        animation.init(this);
    }
    public void start() {
        if (animator.isAlive()) {
            animator.resume();
        } else {
            animator.start();
            suspendRender();
        }
    }
    public void stop() {
        animator.suspend();
    }
    public void destroy() {
        animator.stop();
    }
    public void run() {
        while (true) {
            repaint();
            Thread.sleep(500); // sleep for some time
            animation.advance();
        }
    }
    public void paint(Graphics g) {
        animation.paintFrame(g);
    }
}
```

<47> 도 5는 본 발명에 따라 디스플레이되는, 애플릿을 포함한 마크업 화면의 일 예이다.

<48> 도 5에 도시된 바와 같이, 본 발명에 따른 마크업 화면은 애플릿이 초기화되고 실행이 시작되어 그 첫번째 화면이 디스플레이되면 그에 동기되어 함께 디스플레이되므로 사용자는 종래 애플릿이 초기화되고 실행되는 동안 보여지던 빈 부분(애플릿이 실행되면 보여지기 위한 부

분)을 보지 않아도 된다. 다시 말해, 마크업 문서에 포함되거나 링크된 애플릿과 마크업 문서에 포함된 텍스트/이미지/오디오가 동기되어 동작할 수 있게 되므로 사용자는 보다 편안하게 화면을 시청할 수 있게 된다.

【발명의 효과】

<49> 전술한 바와 같이, 본 발명에 따르면 애플릿이 링크된 마크업 문서의 경우 애플릿과 마크업 문서를 동기시켜 디스플레이할 수 있으므로, 애플릿이 초기화되고 실행 시작을 준비하는 동안 종래 사용자에게 보여지던 빈 화면이 더 이상 보이지 않게 되어, 사용자에게 보다 편안한 화면을 제공할 수 있다.

【특허청구범위】

【청구항 1】

애플릿이 링크된 마크업 문서의 디스플레이 방법에 있어서,

(a) 상기 마크업 문서를 위한 화면 출력 정보의 디스플레이를 지연시키는 단계; 및

(b) 상기 애플릿의 렌더링이 완료되면 지연된 화면 출력 정보와 상기 애플릿 출력을 동기시켜 디스플레이하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 2】

제1항에 있어서,

상기 (a)단계는

상기 마크업 문서를 위한 화면 출력 정보를 버퍼링하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 3】

제1항에 있어서,

상기 (b)단계는

출력 제어 신호에 기초하여 상기 지연된 화면 출력 정보와 상기 애플릿의 첫번째 화면을 위한 애플릿 출력을 디스플레이 장치로 동기시켜 제공하는 단계를 포함하는 것을 특징으로 하는 방법.

【청구항 4】

제1항에 있어서,

상기 애플릿은 마크업 문서의 화면 출력과는 별개의 출력 방식을 가진 프로그램 코드로 이루어진 것임을 특징으로 하는 방법.

【청구항 5】

제4항에 있어서,

상기 출력 제어 신호는 상기 애플릿을 해석하기 위한 애플릿 실행 엔진 또는 상기 마크업 문서를 해석하기 위한 프리젠테이션 엔진으로부터 제공됨을 특징으로 하는 방법.

【청구항 6】

제1항 내지 제3항 중 어느 한 항에 있어서,

상기 (a)단계는

상기 화면 출력 정보로서 텍스트 출력을 포함하고, 이미지 출력 및 오디오 출력 중 적어도 하나를 더 포함하는 것을 특징으로 하는 방법.

【청구항 7】

마크업 문서; 및

상기 마크업 문서에 링크된 애플릿을 포함하고,

상기 애플릿 또는 상기 마크업 문서는 상기 마크업 문서의 디스플레이를 지연시키는 화면 출력 지연 정보를 포함하는 것을 특징으로 하는 정보저장매체.

【청구항 8】

제7항에 있어서,

상기 애플릿은 초기 상태, 실행 상태, 종료 상태 및 삭제 상태 중 어느 하나의 상태에서 동작함을 특징으로 하는 정보저장매체.

【청구항 9】

제7항 또는 제8항에 있어서,

상기 애플릿은 상기 마크업 문서를 디스플레이하기 위한 화면 출력 정보와 상기 애플릿의 출력 정보를 동기화 시키기 위한 디스플레이 지연 정보로서 지연 함수를 포함하는 것을 특징으로 하는 정보저장매체.

【청구항 10】

제9항에 있어서,

상기 애플릿은 상기 마크업 문서를 위한 화면 출력 정보의 디스플레이를 지연시키는 지연 정보로서 지연 함수; 및 상기 애플릿이 초기화되고 실행이 시작되어 그 첫번째 화면의 렌더링이 완료되면 상기 마크업 문서의 디스플레이의 지연을 해제하는 지연 해제 함수를 포함하는 것을 특징으로 하는 정보저장매체.

【청구항 11】

제7항에 있어서,

상기 마크업 문서는 상기 애플릿의 출력 정보와 동기화되어 디스플레이될 것을 요구하는 태그(tag) 또는 속성(attribute) 지시 정보를 포함하는 것을 특징으로 하는 정보저장매체.

【청구항 12】

마크업 문서를 해석하는 프리젠테이션 엔진; 및

상기 마크업 문서에 링크된 애플릿을 해석하는 애플릿 실행 엔진을 포함하고,

상기 프리젠테이션 엔진은

상기 마크업 문서를 위한 화면 출력 정보의 디스플레이를 지연시키고, 상기 애플릿 실행 엔진으로부터 상기 애플릿의 출력에 대한 렌더링이 완료되었음을 알리는 출력 제어 신호가 입력되면, 지연된 화면 출력 정보와 상기 애플릿의 출력을 동기시켜 디스플레이 장치로 출력하는 것을 특징으로 하는 시스템.

【청구항 13】

제12항에 있어서,

상기 프리젠테이션 엔진은

상기 마크업 문서를 위한 화면 출력 정보의 디스플레이를 지연시키기 위해 버퍼링하는 버퍼를 포함하는 것을 특징으로 하는 시스템.

【청구항 14】

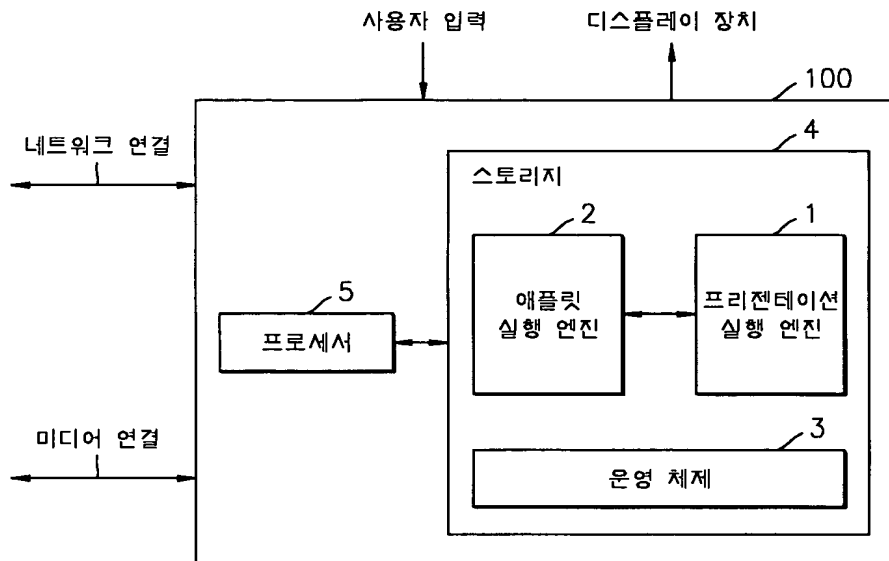
제12항에 있어서,

상기 프리젠테이션 엔진은

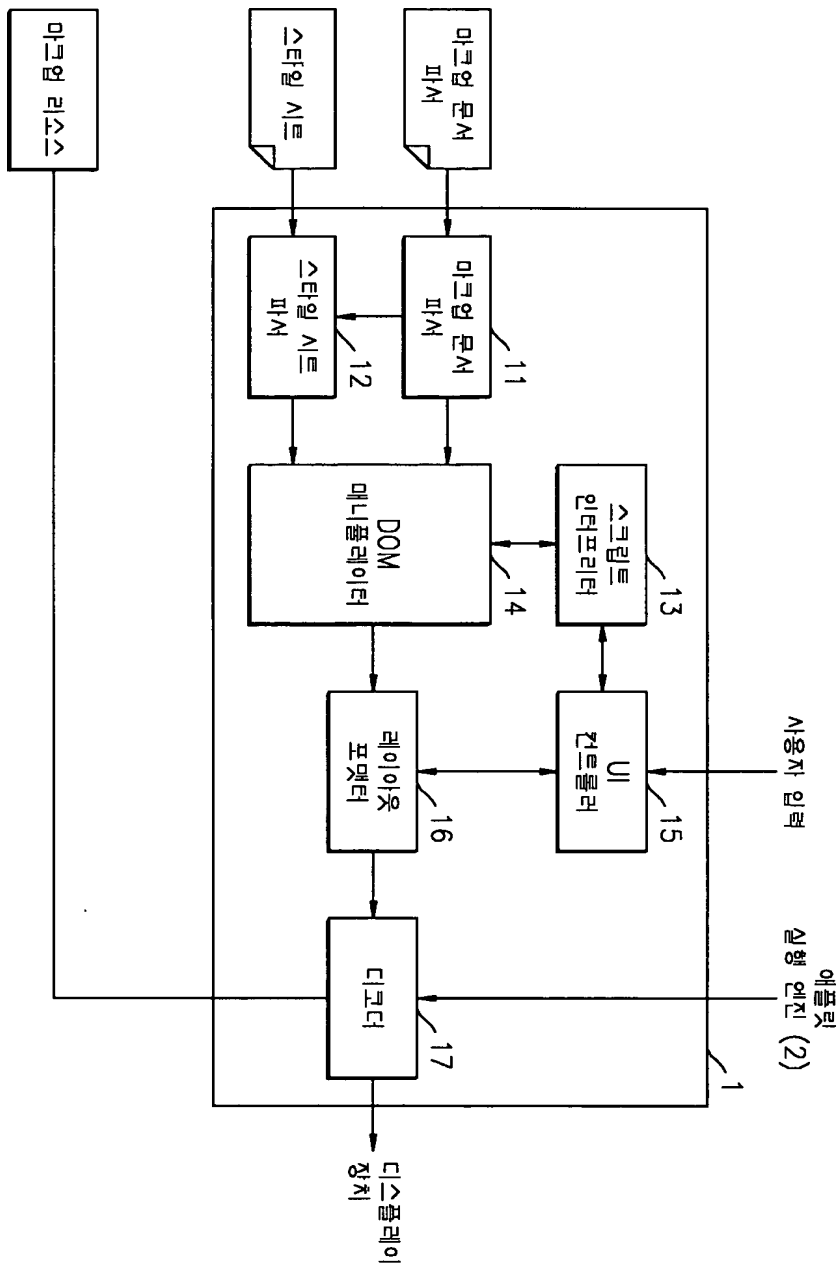
오디오 출력을 버퍼링하기 위한 오디오 버퍼와 비디오 출력을 버퍼링하기 위한 비디오 버퍼를 구비함을 특징으로 하는 시스템.

【도면】

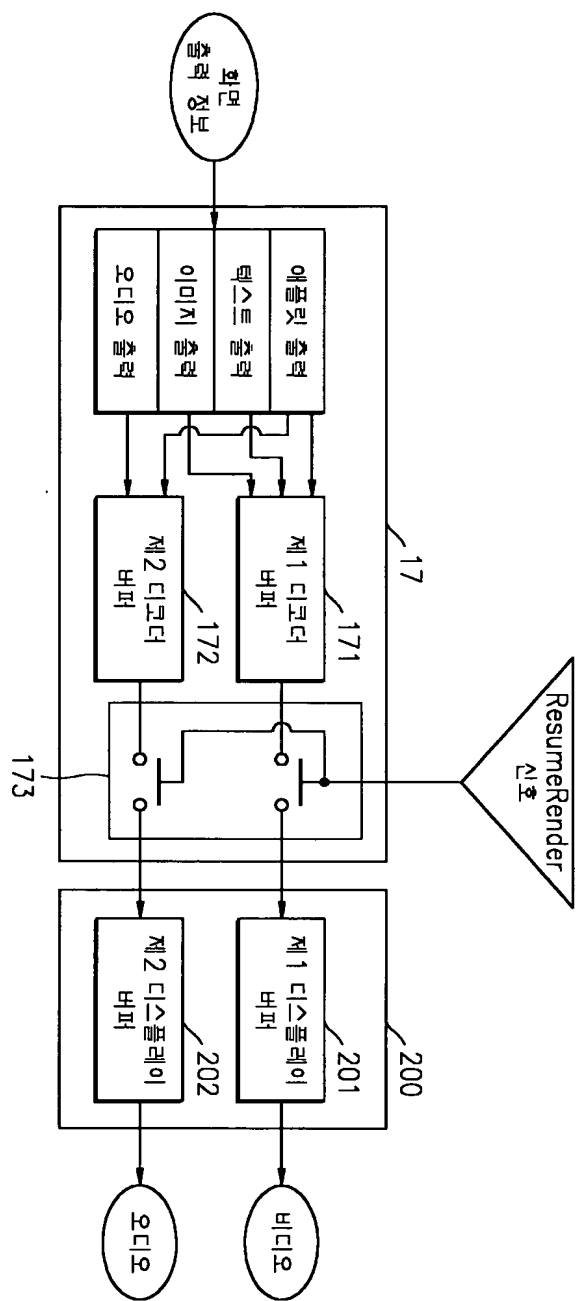
•【도 1】



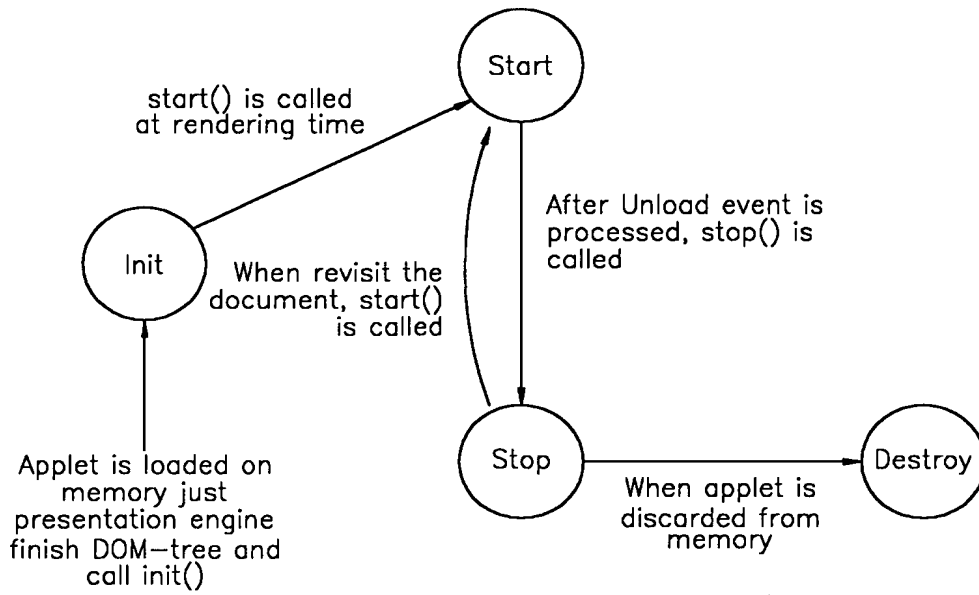
【도 2】



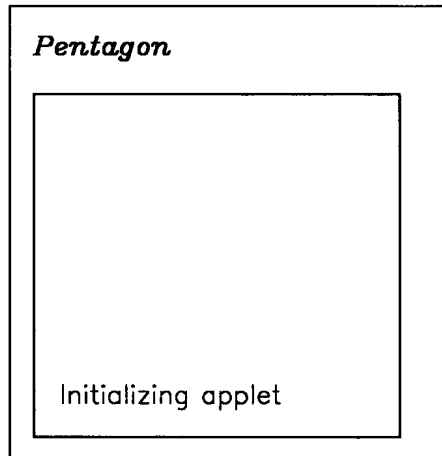
【도 3】



【도 4】



【도 5】



【도 6】

